Debug Methodology for the McKinley Processor

Doug Josephson
Hewlett Packard

Steve Poehlman
Intel

Vincent Govan
Hewlett Packard

Purpose of Work

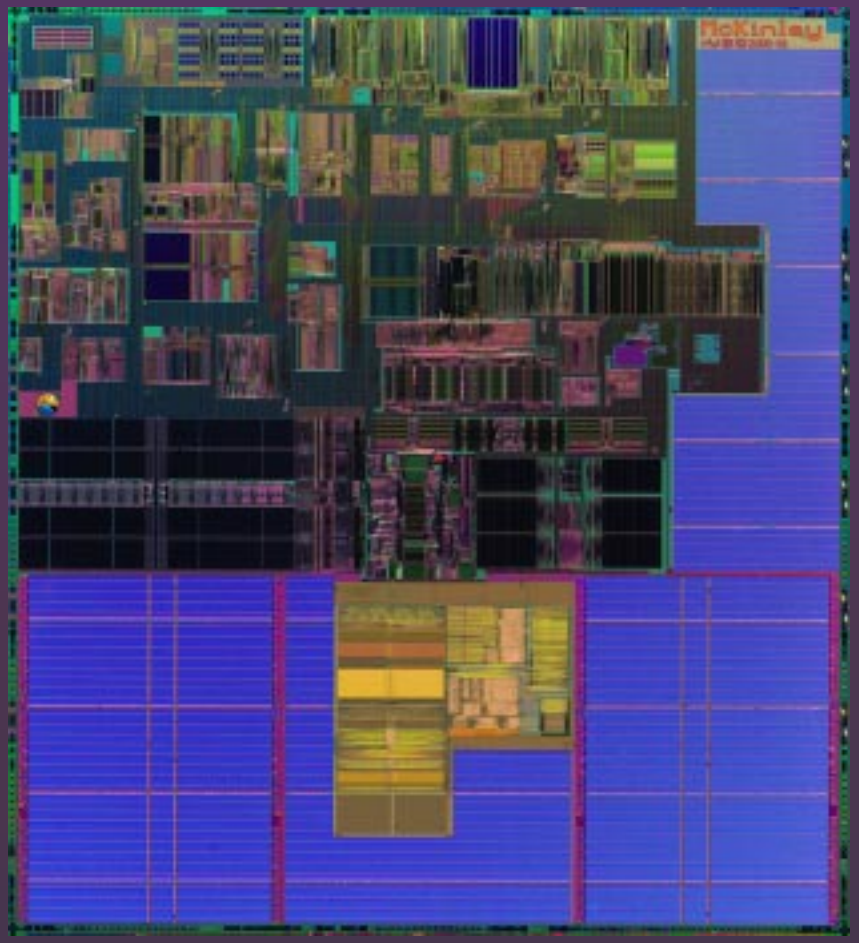
- Enable McKinley processor debug
- Deliver a robust, high quality design
- Investigate new debug techniques
- Increase effectiveness of debug
  - Decrease bug turn-around time
  - Increase debugger productivity
- Learn for the future

Presentation Overview

- McKinley overview
- Functional and electrical verification
- Tester and system debug
- McKinley debug features
- Debug example
- Results/conclusion

McKinley Overview

- Intel/HP designed
- New microarchitecture
- 221 million FETs
- 1 GHz operation
- 8 stage pipeline
- 6 wide super scalar
- 11 issue ports
- 3 levels of cache
- 400 MT/s bus



Design Verification

- Functional
  - "Is the design logically correct?"
  - Performed at a "safe" operating point
- Electrical
  - "Is the design robust?"
  - Frequency, voltage, temperature, process
  - "Shmoo to failure"
  - Tests the design to well beyond specs
- Random/focused techniques are used

Tester / System Debug

Tester		System
Expensive	⟷	Cheap
Per pin control	⟷	Hard to control pins
Deterministic	⟷	Non-deterministic
Hard vector generation	⟷	Easy vector generation
Enables easy debug	⟷	Complicated debug

Use of both is complementary and necessary!

McKinley Debug Features

- Debug unit
- Probe mode
- Scan debug
- Clock manipulation
- Programmable clock skewing
- IEEE 1149.1 access via code
- Timing on the fly

Traditional Debug Features

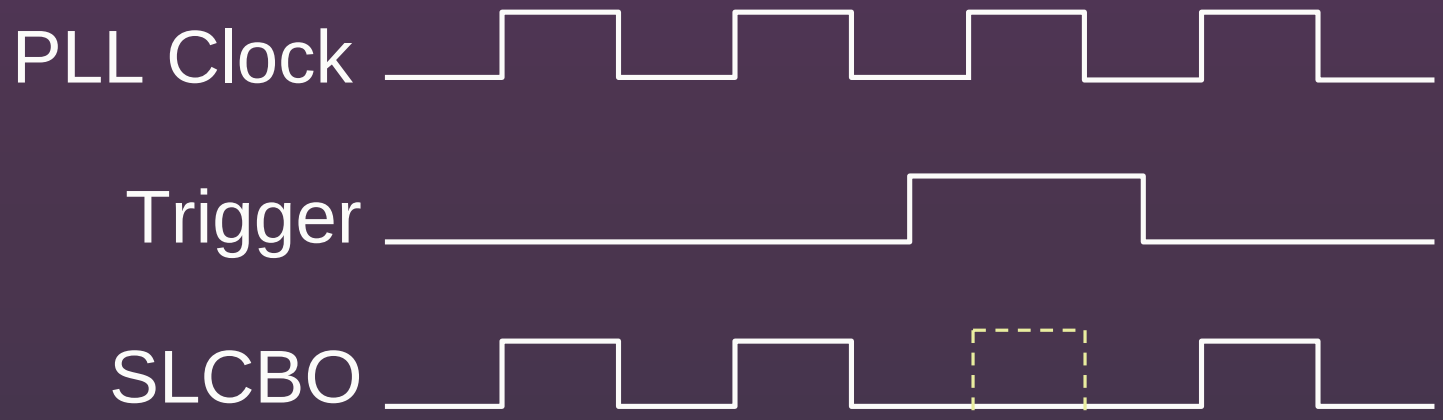
- Debug unit
  - Observes chip state and signals
  - Generates trigger events for debugging
  - Very complex triggering is possible
- Probe mode
  - Allows debugger to probe internal state
  - Triggered stall allows control of processor
  - Instructions can be single-stepped via scan

Scan Debug

- Sample on the fly
  - Non-intrusive method of observing signals
  - Access to 24,000 critical signals in design
  - Vital for debugging rare failures
- Full scan
  - Destructive method of observing signals
  - Access to 136,000 signals in design
  - Excellent capability to pinpoint failures

Digital Clock Manipulation

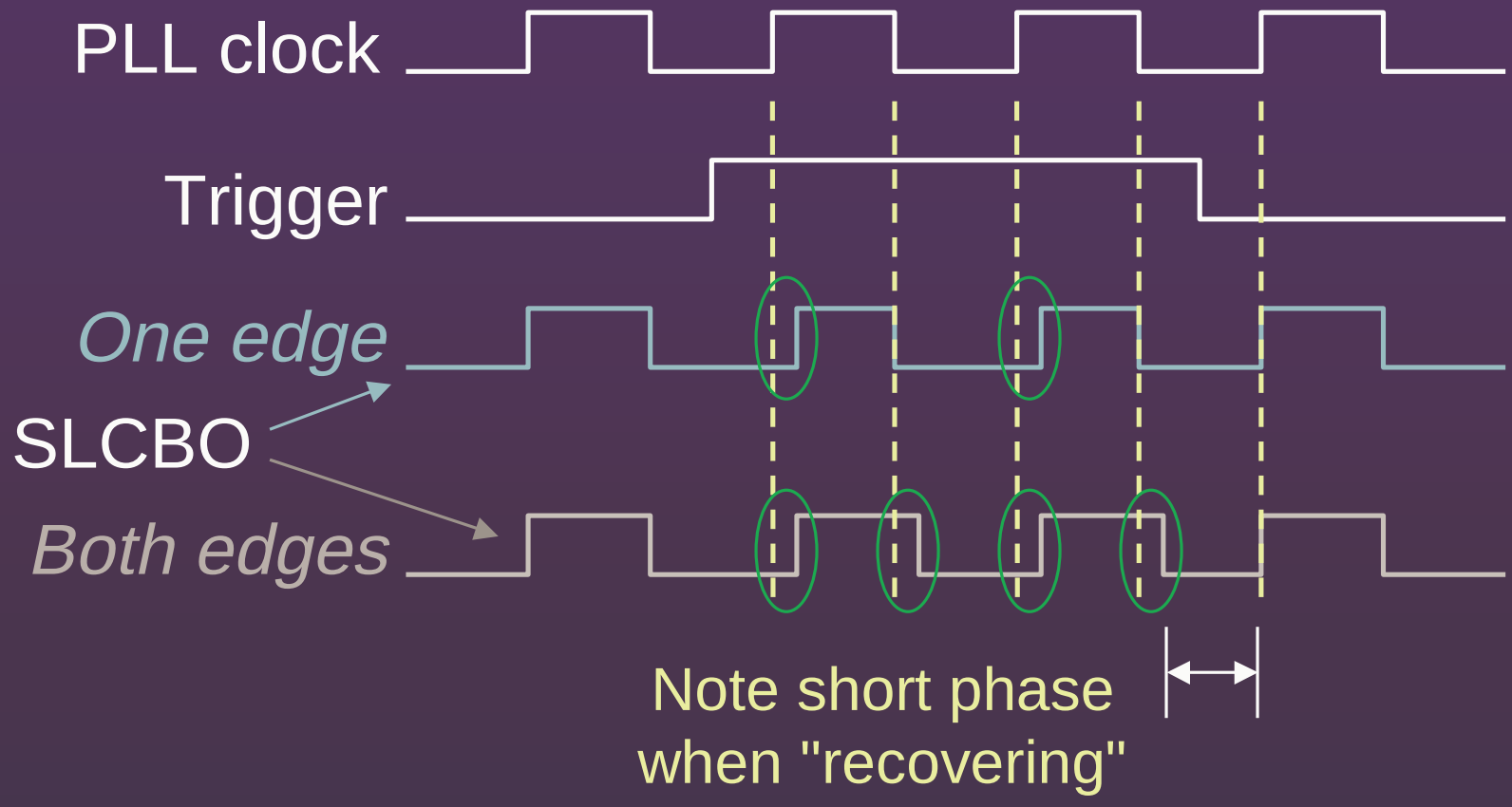
- "Skips" clocks based on triggers
- One or more cycles possible
- Easy to use
- Problematic with a synchronous bus



Analog Clock Manipulation

- Delays clock edges based on triggers
- Delays either or both rising/falling edges
- Can delay up to 120 ps in typical case
- Advantages
  - Bus remains synchronized to core clock
  - Avoids issues with bypass cap recharge
- Disadvantage
  - Changing back to master PLL clock

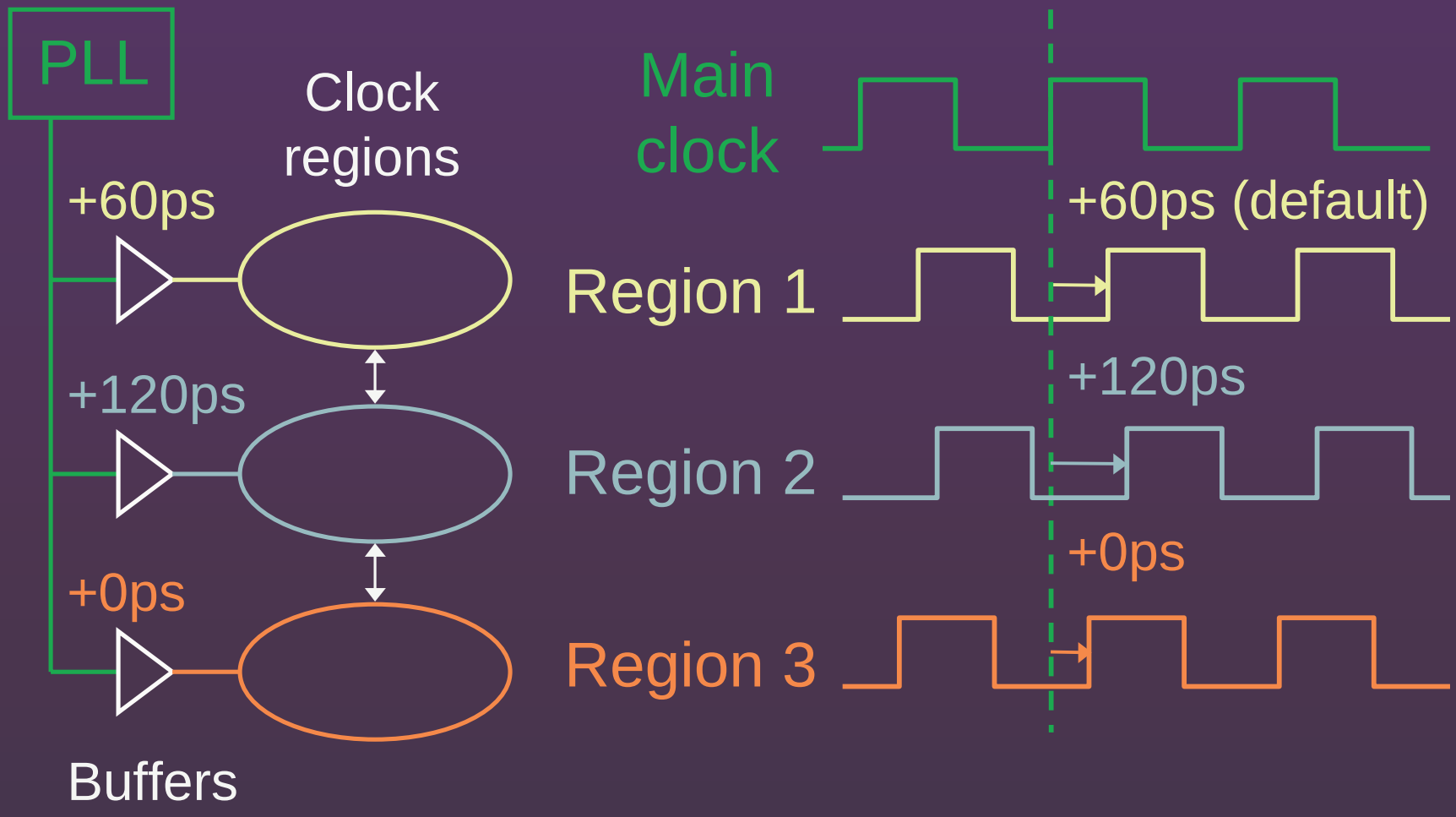
Two Manipulation Examples



Programmable Clock Skewing

- Individual control of 33 clock regions
- Regions can be skewed up to 150 ps
- Can be used in normal operation
- Advantages and disadvantages
  - Can isolate interblock timing issues
  - Can be used as a "workaround" for bugs
  - Can break other paths
  - Introduces additional skew in clock tree

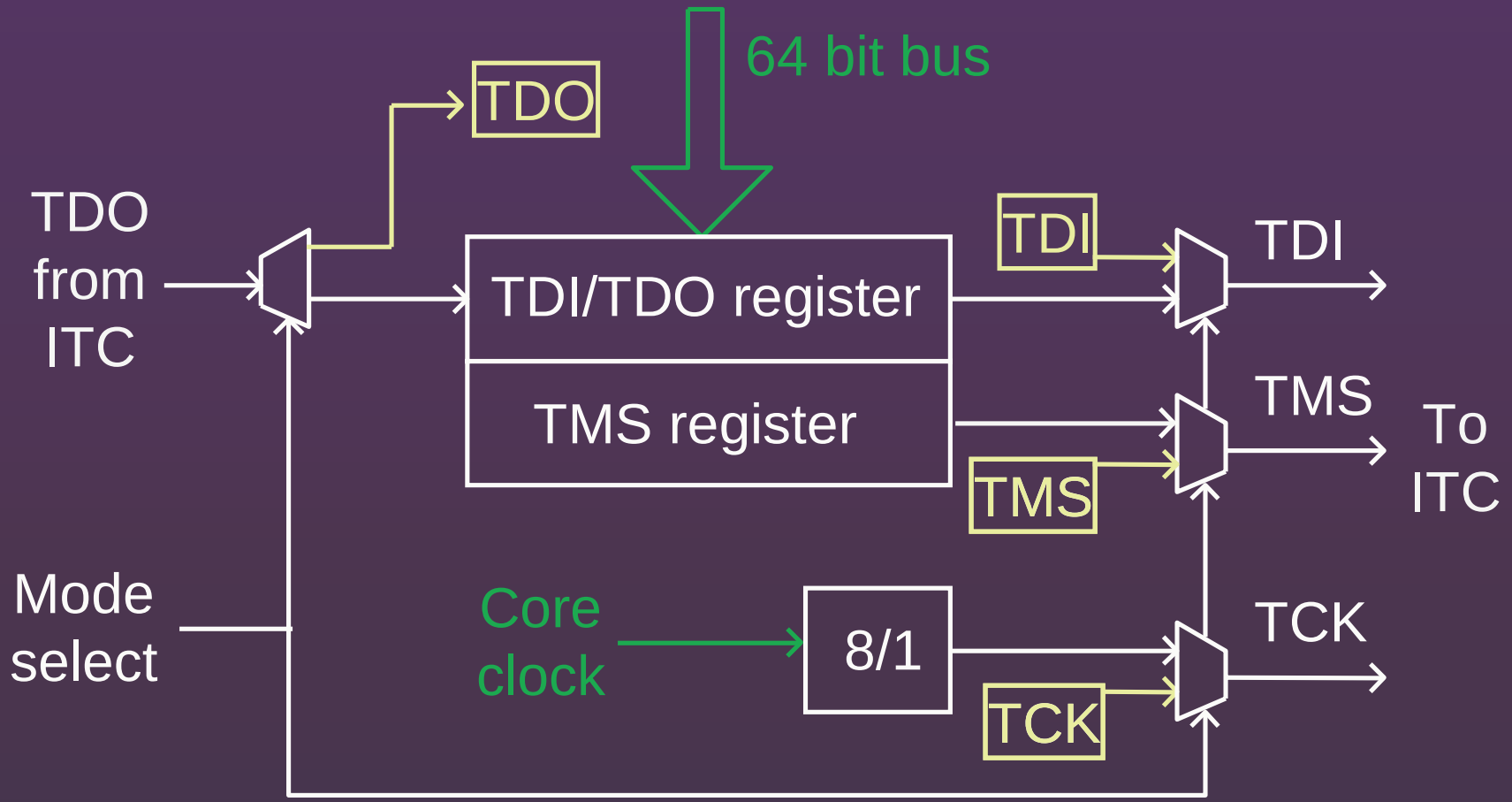
Clock Skewing Example



IEEE 1149.1 Access Via Code

- Test access with no external hardware
- Allows code to control test features
- Advantages and disadvantages
  - Low overhead access to test capability
  - No external hardware required
  - Chip must be functional
  - No access to destructive scan
  - All other test features are usable

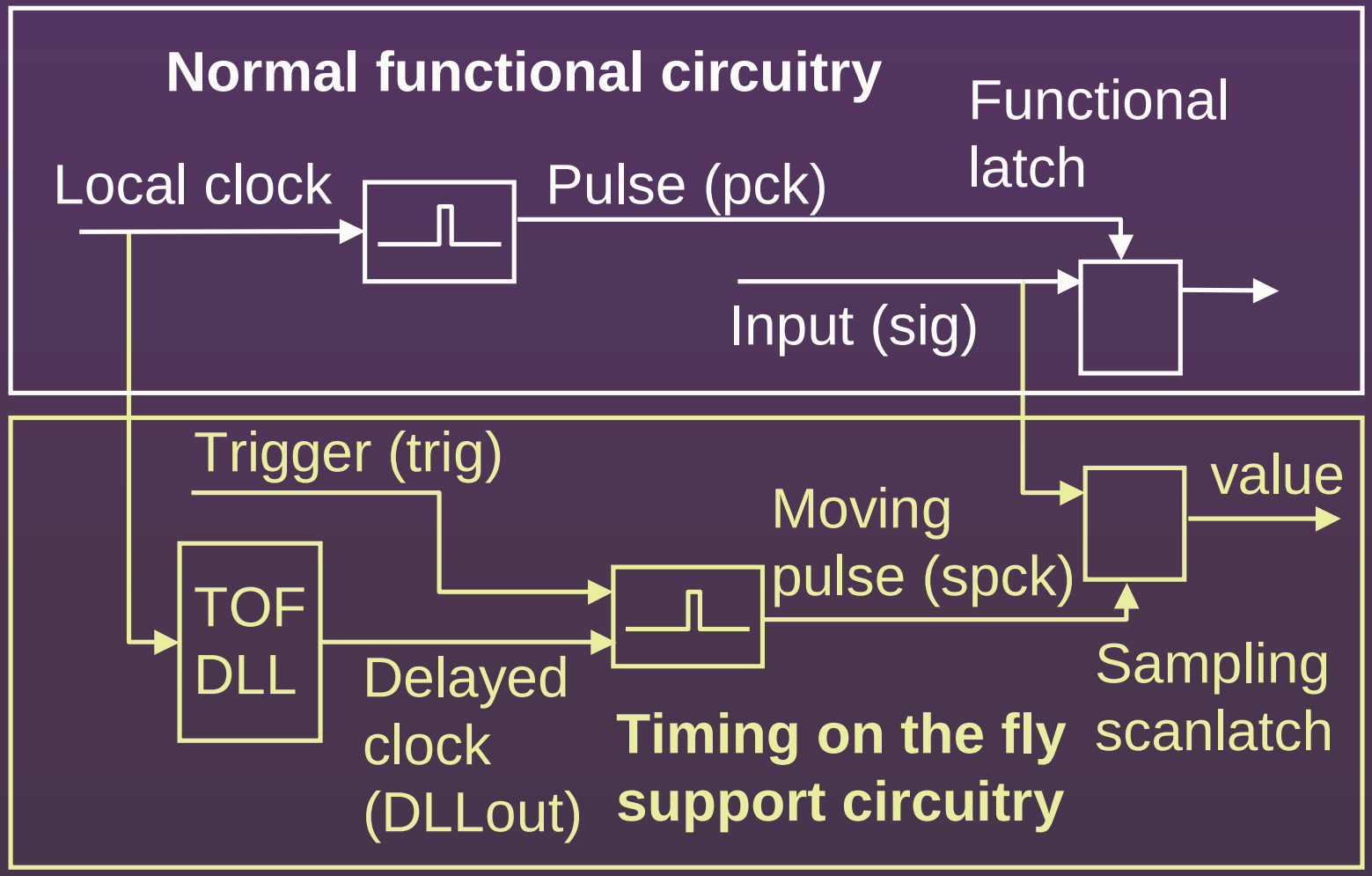
Implementation Example



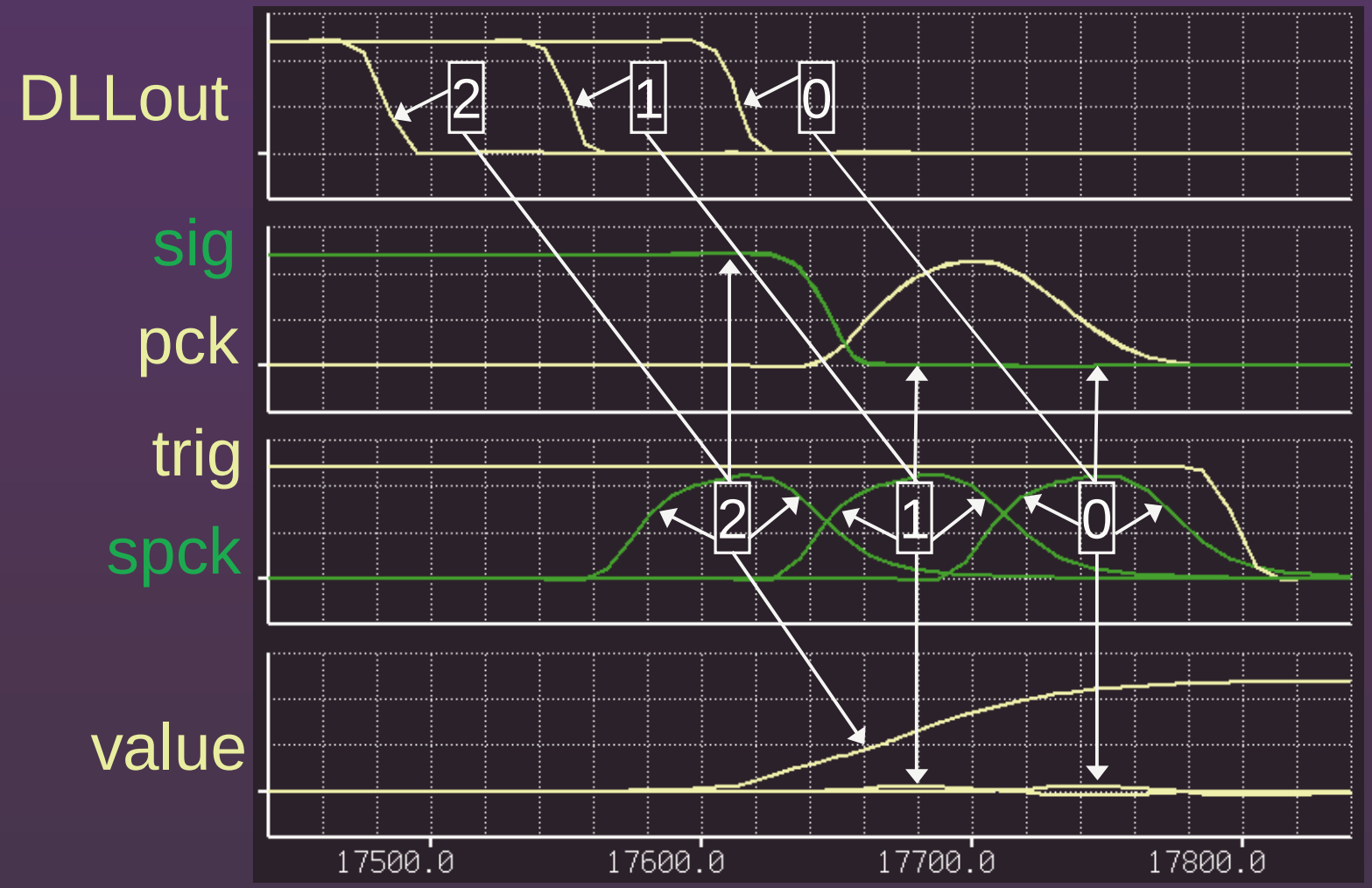
Timing on the Fly

- Method of determining internal timing
- Extension of sample on the fly
- Resolution of approximately 60 ps
- Captures approximately 1000 signals
- Useful on electrically critical signals
  - Known timing problems
  - "Tricky" signals known to be risky
- Currently under evaluation

Timing on the Fly Diagram

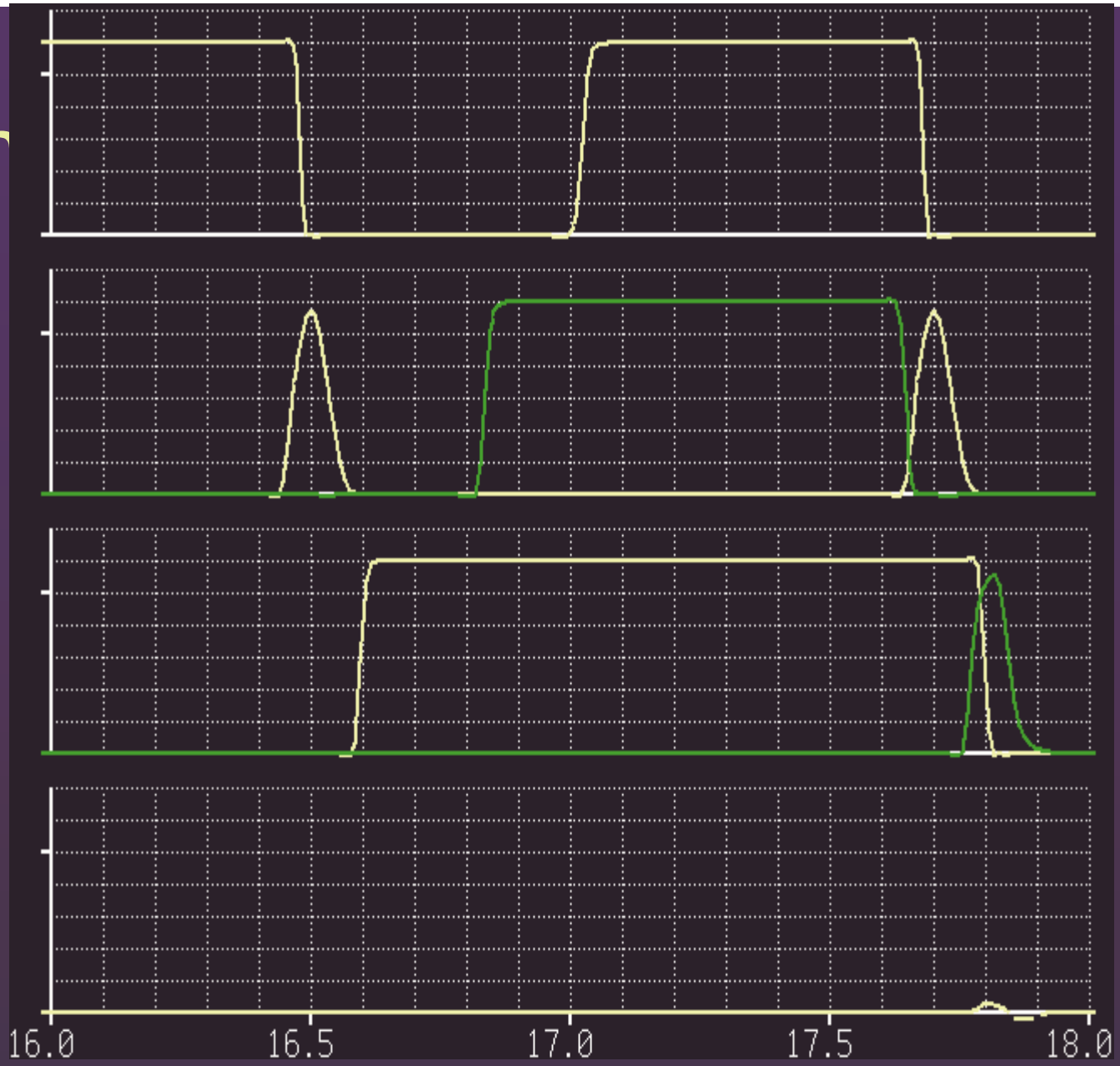


Timing on the Fly SPICE



Time (20 ps/div)

Timir


DLLout
sig
pck
trig
spck
value
16.0
16.5
17.0
17.5
18.0
Time (200 ps/div)

Timing on the Fly Operation

Actual LVP Delays

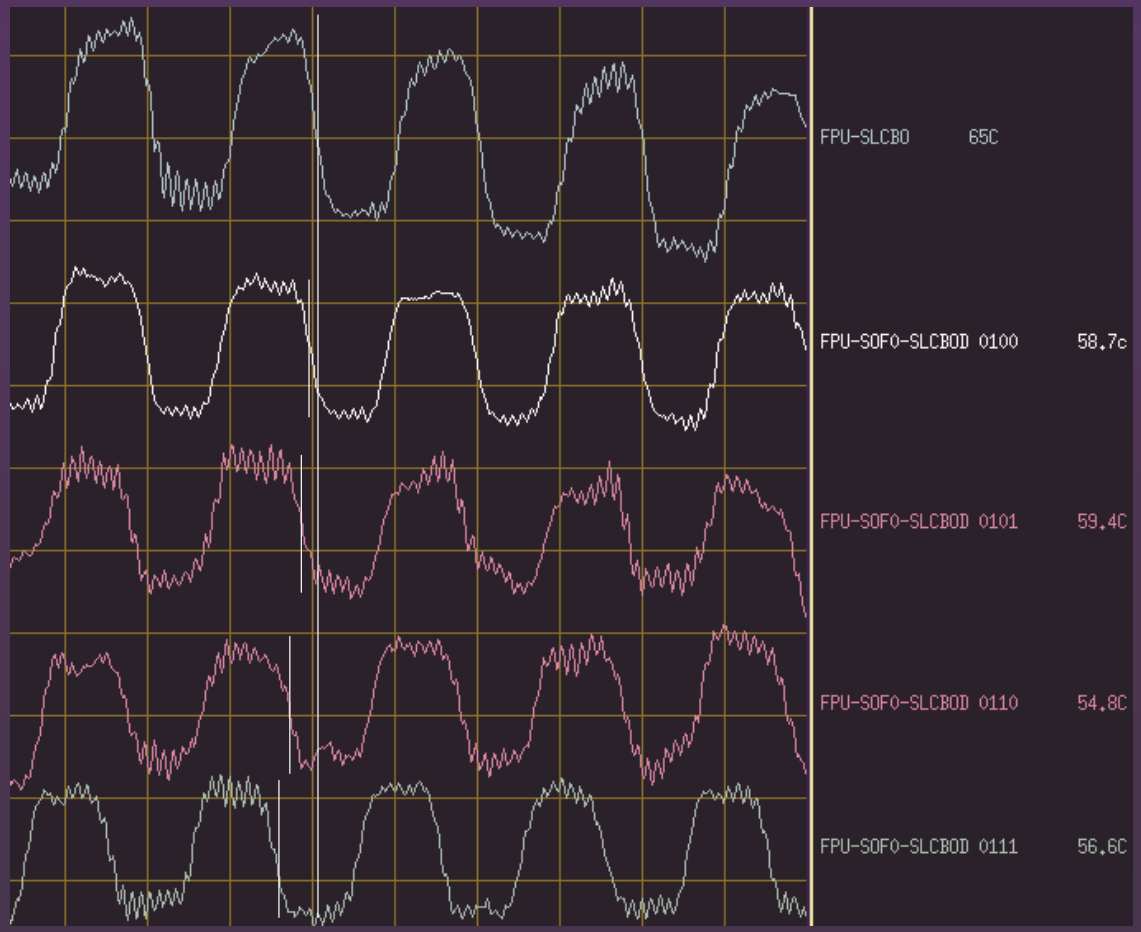
DLL input clock

-55 ps

-83 ps

-167 ps

-250 ps



500 ps/div

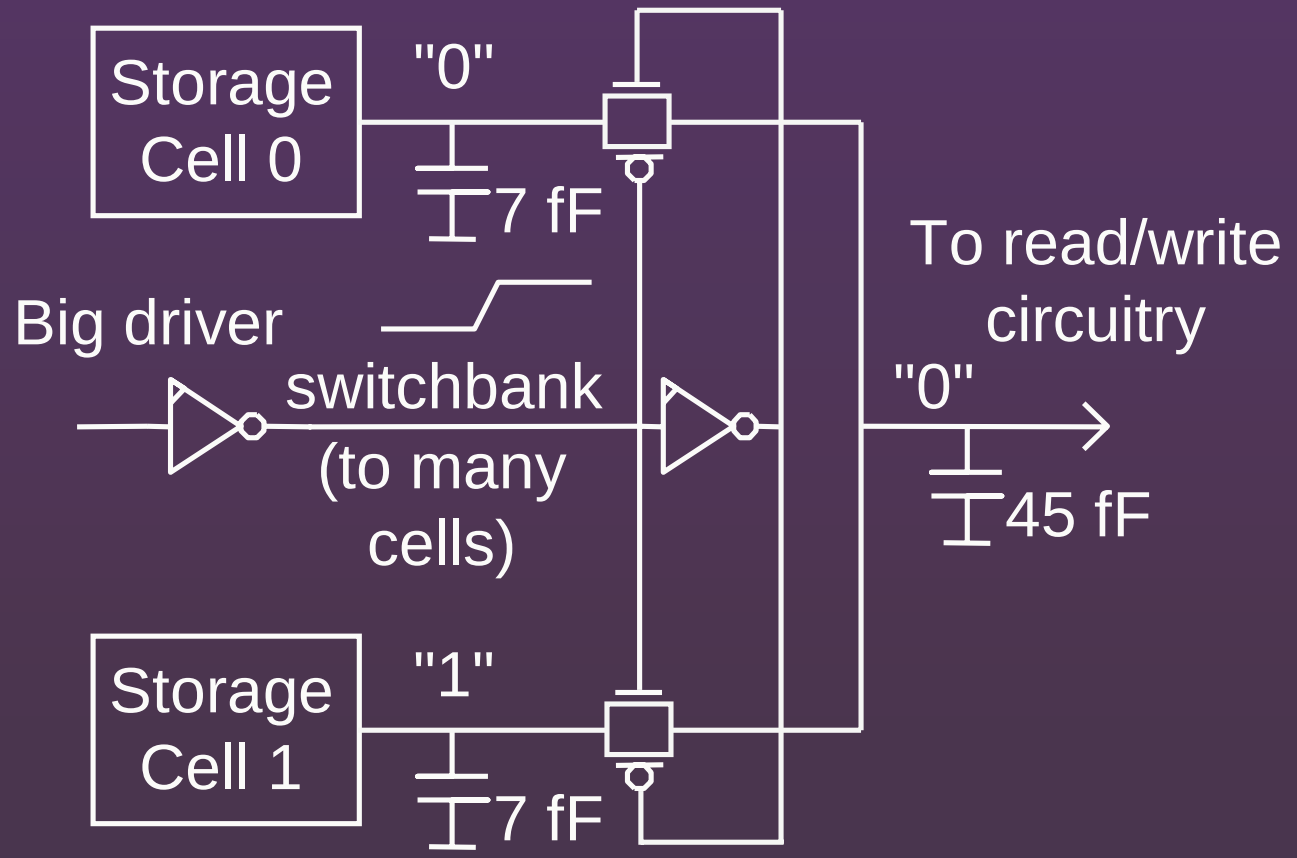
McKinley Electrical Debug Flow

- Failures observed on tester/system
- Experiments performed
- Clock manipulation to find critical cycle
- Scan data collected to identify failure
  - Sample on the fly to narrow in on failure
  - Stop clock and do full scan
- Failure hypothesis generated
  - SPICE and/or LVP performed
  - FIB experiments performed

Bug Example

- Failure to boot system at high voltage
  - Code narrowed down to one instruction
  - Took scan data, did code experiments
- Performed SPICE based on failure
  - This showed a charge sharing problem
  - FIB experiment done to cut driver size
  - Fixed charge share
  - Introduced a speedpath (expected)
  - Fixed entire problem in FET release

Bug Schematic



Results

- Debug features have worked very well
  - Sample on the fly and full scan vital
  - Clock manipulation very useful
  - Debug time substantially reduced
  - New features still being characterized
- Opportunities identified for future
  - Extend capability to gather scan traces
  - More advanced clock manipulation

Conclusion

- McKinley debug is complex
  - New architecture and implementation
  - New circuit design techniques
  - Challenging performance goals
  - Joint design and debug teams
- Results have exceeded expectations
- We acknowledge our colleagues for their exceptional work!